CLAIMS

What is claimed is:

1. A computer system comprising:

a plurality of groups of data in a common software language, wherein each group of data comprises;

a first group of data modeling a plurality of components and a plurality of interactions between said plurality of components;

a second group of data modeling a software load of said plurality of components and said plurality of interactions between said plurality of components; and

a first software code interacting with said plurality of groups of data for qualifying said second subset of data as defining a valid combination of said plurality of components.

2. The computer system according to Claim 1, wherein said first group of data comprises:

a first subset of data for modeling said plurality of interactions; and

a second subset of data for modeling said plurality of components.

3. The computer system according to Claim 1, wherein said first group of data comprises:

a first subset of data for modeling software components; and

a second subset of data for modeling hardware components.

4. The computer system according to Claim 1, wherein said first software code interacts with said plurality of groups of data utilizing a first set of predefined rules.

5. The computer system according to Claim 4, wherein:

said first group of data comprising version-identifying data for said plurality of interactions and version range identifying data related to said plurality of components; and

each of said first set of predefined rules comprise the version of a particular interaction lying within said version range of a first component and a second component cooperating through said particular interaction.

6. The computer system according to Claim 5, wherein said first group of data defines a plurality of direction interaction, wherein each direction interaction is between a server component providing one of said plurality of interfaces and a client component utilizing said one of said plurality of interfaces.

7. The computer system according to Claim 6, wherein:

said first set of data comprises generic "provide" and "use" entities each having a version range identifying attribute, with instance of said "provide" and "use" entities attached to said server component and client component with values of said version range identifying attribute, respectively; and

said first predefined rules comprise:

initially verifying that said first group of data comprise an interaction for each version value in each instantiated version range; and for a given pair of sever component and client component, verifying that the intersection of their respective “provide” and “use” version ranges is not nil.

8. The computer system according to Claim 1:

wherein said plurality of groups for data comprise a plurality of generic entities for said plurality of components, said plurality of interactions and a software load;

wherein said entities comprise predefined numbers; and

wherein said first group of data and said second group of data utilizes such generic entities for representing said plurality of components, said plurality of interactions and said software load.

9. The computer system according to Claim 1, wherein said plurality of generic entities comprise attributes and sub-entities.

10. The computer system according to Claim 1, wherein said plurality of groups of data are stored in accordance with a tree structure.

11. The computer system according to Claim 1, wherein said plurality of groups of data further comprises a third group of data modeling configuration values for loading a software load as defined in said second group of data.

12. The computer system according to Claim 11, further comprising a second software code, wherein said second software code interacts with said third group of data for verifying that said third group of data define a set of required configuration values.

13. The computer system according to Claim 12, wherein:

said plurality of groups of data comprise generic entities for interactions, with a configuration attribute having an attribute statement; and

said second software code reverts to said first group of data for determining said configuration attribute.

14. The computer system according to Claim 13, wherein:

said group of data also comprises generic entities for components, software load and configuration values;

said generic entities having a configuration value attribute;

said first, second and third group of data comprise instances of said generic entities for representing said components, said software load and said configuration values; and

said second software code reverts from said third group of data through said second group of data to said first group of data, for determining whether a configuration value is present.

15. The computer system according to Claim 13, wherein:

said generic entities each comprise a final configuration attribute for at least one of said components, said interactions and said software load; and

said second software code is responsive to finding an attribute value qualified as final, and nevertheless finding an attribute value downstream in the sense: interaction, component, software load, configuration value for entering an error processing mode.

16. The computer system according to Claim 11, further comprising:

said plurality of groups of data further comprises:

a fourth set of data comprising platform update data for designating a configuration update having an update level; and

a thirds software code determines if a transition to said configuration update is authorized.

17. The computer system according to Claim 16, wherein said third software code further determines if a rollback from said configuration update is authorized.

18. The computer system according to Claim 16, wherein said plurality of groups of data further comprises a current load.

19. The computer system according to Claim 18, wherein said current load is at least partially represented in the form of directory data.

20. The computer system according to Claim 16, further comprising a fourth software code for implementing a new configuration value in a platform based on said plurality of groups of data.

21. The computer system according to Claim 16, wherein said fourth data further comprises a new software load.

22. The computer system according to Claim 21, further comprising a fifth software code, capable of building a software load image implementing said new software load based on said plurality of groups of data and on a package file.

23. The computer system according to Claim 1, further comprising a sixth software code capable of indicating a state of a component from said plurality of groups of data in response to a request.

24. The computer system according to Claim 23, further comprising a management service code utilizing said sixth software code.

25. The computer system according to Claim 24, wherein said management service code comprises local management service code.

26. The computer system according to Claim 24, wherein said management service code comprises a management agent enabling remove management.

27. The computer system according to Claim 1,

wherein said plurality of groups of data is defined from generic entities and from predefined instances of said generic entities;

wherein said group of data is represented in a tree structure; and

wherein said tree structure includes a system section comprising said generic entities and said predefined instance of such generic entities.

28. The computer system according to Claim 1, wherein said first group of data comprises data associated with referencing instances and component assignments.

29. The computer system according to Claim 1, wherein said common software language is a markup language.

30. A method of modeling a computer system, comprising:

generating a plurality of sets of data comprising;

a first set of data for modeling a plurality of components;

a second set of data for modeling a plurality of interactions between said plurality of components;

a third set of data for modeling a plurality of software loads as a function of said first and second set of data; and

qualifying said third set of data as defining a valid combination of said plurality of components.

31. The method according to Claim 30, wherein said first set of data comprises:

a first subset of data for modeling software components of said plurality of components; and

a second subset of data for modeling hardware components of said plurality of components.

32. The method according to Claim 30, wherein said qualifying said third set of data is based upon a first set of predefined rules.

33. The method according to Claim 32, wherein:

said first set of data comprises a version range identifying data for each of said plurality of components;

said second set of data comprises a version identifying data for each of said plurality of interactions; and

each of said first set of predefined rules comprises a version of a particular interaction lying within the version range of a first and a second one of said plurality of component cooperating through said particular interaction.

34. The method according to Claim 30, wherein:

said plurality of sets of data comprises a plurality of generic entities for said plurality of components, said plurality of interactions and said plurality of software loads;

said generic entities comprise predefined numbers; and

said plurality of set of data utilize said generic entities for representing said plurality of components, said plurality of interactions and said plurality of software loads.

35. A computer-readable medium containing a plurality of instructions which when executed cause a computing device to implement a method of modeling a computer system, comprising:

generating a searchable tree data structure comprising;

a first set of data for modeling a plurality of components;

a second set of data for modeling a plurality of interactions between said plurality of components;

a third set of data for modeling a plurality of software loads as a function of said first and second set of data; and

qualifying said third set of data as defining a valid combination of said plurality of components.

36. The computer-readable medium according to Claim 35, wherein generating said tree data structure further comprises a fourth set of data for modeling configuration values for loading a plurality of software as a function of said third set of data.

37. The computer-readable medium according to Claim 35, further comprising:

said generating said searchable tree data structure further comprises a fourth set of data comprising platform configuration update data; and

determining if a transition to a particular update level is authorized as a function of said fourth set of data.

38. The computer-readable medium according to Claim 37, further comprising determining if a rollback from said particular update level is authorized as a function of said fourth set of data.

39. The computer-readable medium according to Claim 35, further comprising determining a state of a particular component as a function of said searchable tree data structure.